Spatially Prioritized and Persistent Text Detection and Decoding

Spatially Prioritized and Persistent Text Detection and Decoding

Hsueh-Cheng Wang, Yafim Landa, Maurice Fallon, and Seth Teller
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, MA 02139 USA

***Abstract*—We show how to exploit temporal and spatial coherence to achieve efficient and effective text detection and decoding for a sensor suite moving through an environment in which text occurs at a variety of locations, scales and orientations with respect to the observer. Our method uses simultaneous localization and mapping (SLAM) to extract planar "tiles" representing scene surfaces. It then fuses multiple observations of each tile, captured from different observer poses, using homography transformations. Text is detected using Discrete Cosine Transform (DCT) and Maximally Stable Extremal Regions (MSER) methods; MSER enables fusion of multiple observations of blurry text regions in a component tree. The observations from SLAM and MSER are then decoded by an Optical Character Recognition (OCR) engine. The decoded characters are then clustered into character blocks to obtain an MLE word configuration.**

This paper's contributions include: 1) spatiotemporal fusion of tile observations via SLAM, prior to inspection, thereby improving the quality of the input data; and 2) combination of multiple noisy text observations into a single higher-confidence estimate of environmental text.

***Keywords*—SLAM, Text Detection, Video OCR, Multiple Frame Integration, DCT, MSER, Lexicon, Language Model**

I. Introduction

Information about environmental text is useful in many task domains. Examples of outdoor text include house numbers and traffic and informational signage; indoor text arises in building directories, aisle guidance signs, office numbers, and nameplates. Given sensor observations of the surroundings we wish to efficiently and effectively detect and decode text for use by mobile robots or by people (e.g. the blind or visually impaired). A key design goal is to develop text extraction method which is fast enough to support real-time decision-making, e.g. navigation plans for robots and generation of navigation cues for people.

A. End-to-End Text Spotting in Natural Scenes

Aspects of end-to-end word spotting have been explored previously. Batch methods for Optical Character Recognition (OCR) have long existed. In a real-time setting, however, resource constraints dictate that text decoding should occur only in regions that are likely to contain text. Thus, efficient text detection methods are needed. Chen and Yuille [3] trained a strong classifier using AdaBoost to identify text regions, and used commercial OCR software for text decoding.

Neumann and Matas [19], [20], [21] used Maximally Stable Extremal Region (MSER) [15] detection and trained a classifier to separate characters from non-characters using several shape-based features, including aspect ratio, compactness, and convex hull ratio. They reported an average run time of 0.3 s on an 800×600 image, achieving recall of 64.7% in the ICDAR 2011 dataset [14] and 32.9% in the SVT dataset [31].

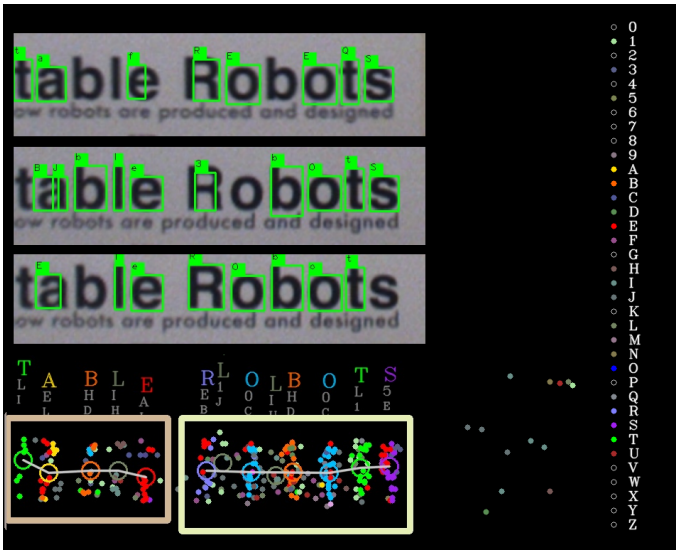

Fig. 1. Our approach incorporates Simultaneous Localization and Mapping (SLAM) to combine multiple noisy text observations for further analysis. Top left: three cropped tile observations with decoded characters. Bottom left: the spatial distribution of decoded characters from all observations (each dot is a decoded character). A clustering is first used to group decoded characters; each group is shown as a circle, positioned at the centroid of the decoded characters. A second clustering step merges each group (circle) into a word candidate, represented as a rectangle. Next, an optimal word configuration is obtained, e.g., two groups of "L" are excluded, shown by line segments connecting circles. The final outputs "TABLE" and "ROBOTS" (from source text "Printable Robots") are the optimal character sequences computed using a language model. Right: a legend - each dot represents one character (case-insensitive).

Wang and colleagues [31], [30] described a character detector using Histograms of Oriented Gradient (HOG) features or Random Ferns, which given a word lexicon can obtain an optimal word configuration. They reported computation times of 15 seconds on average to process an 800×1200 image. Their lexicon driven method — combining the ABBYY FineReader OCR engine and a state-of-the-art text detection algorithm (Stroke Width Transform (SWT) [5]) — outperformed the method using ABBYY alone.

The open-source OCR engine Tesseract [28], [29] has some appealing features, such as line finding, baseline fitting, joined character chopping, and broken character association.

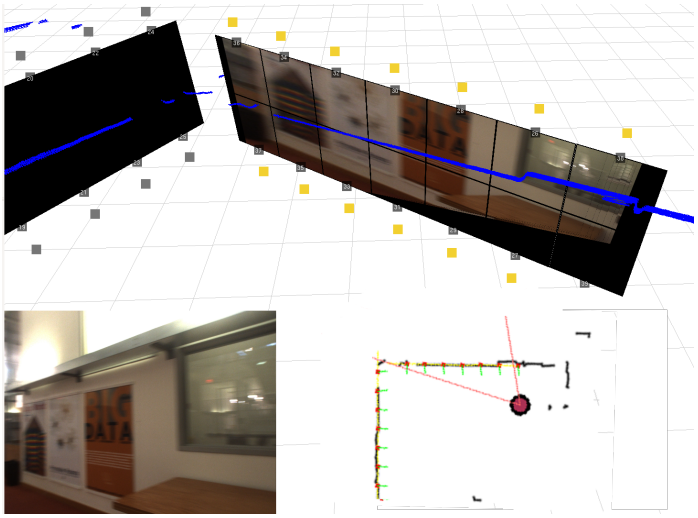

Fig. 2. Top: Visualization of a 3D environment. Each yellow or black label represents an 1×1 meter tile. The yellow ones are in camera field of view, and the black ones are discovered by LIDAR, but not by camera. Bottom left: A camera frame. Bottom right: Map generated by the SLAM module (black lines) with generated tiles overlaid (origins in red; normals in green).

Although its accuracy was not as high as that of some other commercial OCR engines [31], it has been widely used in many studies.

B. Challenges

We address the problem of extracting useful environmental text from the datastream produced by a body-worn sensor suite. We wish to extract text quickly enough to support real-time uses such as navigation (e.g., the user seeks a numbered room in an office or hotel), shopping (e.g., the user seeks a particular aisle or product), or gallery visits (e.g. the user wants notification and decoding of labels positioned on the walls and floors, or overhead).

To achieve real-time notifications given current network infrastructure, the processing should be performed on-board (i.e., by hardware local to the user), rather than in the cloud, and in a way that exploits spatiotemporal coherence (i.e. the similarity of data available now to data available in the recent past). First, the user often needs a response in real time, ruling out the use of intermittent or high-latency network connections. Second, the task involves large amounts of data arising from observations of the user's entire field of view at a resolution sufficient for text detection. This rules out reliance on a relatively low-bandwidth network connection. Moreover, in 2013 one cannot analyze a full field of view of high-resolution pixels in real-time using hardware that would be reasonable to carry on one's body (say, a quad- or eight-core laptop). We investigated what useful version of the problem could be solved with wearable hardware, and designed the system to inspect, and extract text from, only those portions of the surroundings that are *newly visible*.

Existing work has incorporated scene text in robotics [25] and assistive technologies for visually impaired or blind people [32]. Unlike scene text in images observed by a stationary camera, text observed by a moving camera will generally be

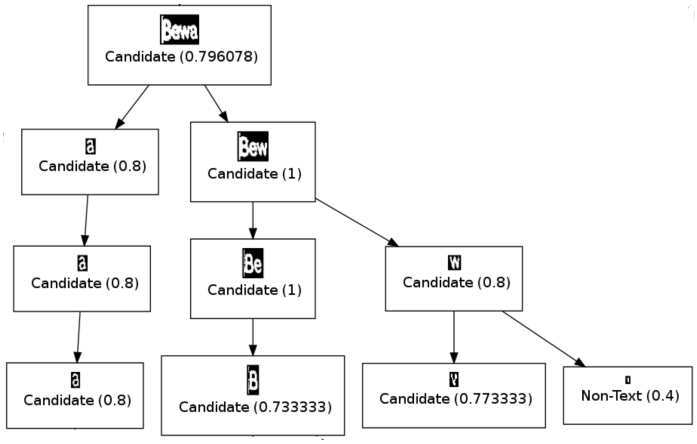

Fig. 3. MSER component tree. Each node was classified as (potential) text, or as non-text, based on shape descriptors including compactness, eccentricity, and the number of outer boundary inflexion points.

subject to motion blur or limited depth of field (i.e. lack of focus). Blurry and/or low-contrast images make it challenging to detect and decode text. Neither increasing sensor resolution, nor increasing CPU bandwidth, are likely to enable text detection alone; instead, improved methods are required.

For blurry or degraded images in video frames, multi-frame integration has been applied for *stationary* text [27], [13], [9], e.g., captions in digital news, and implemented for text enhancement at pixel or sub-pixel level (see [12]). However, additional registration and tracking are required for text in 3D scenes in video imagery [18].

II. The Proposed Method

SLAM has long been a core focus of the robotics community. On-board sensors such as cameras or laser range scanners (LIDARs) enable accurate egomotion estimation with respect to a map of the surroundings, derived on-line. Large-scale, accurate LIDAR-based SLAM maps can now be generated in real time for a substantial class of indoor environments. Incremental scan-matching and sensor fusion methods have been proposed by a number of researchers [23], [1], [6]. We incorporate SLAM-based extraction of 1m $\times$ 1m "tiles" to improve text-spotting performance.

Our system uses SLAM to discover newly visible vertical tiles (Fig. 2), along with distance and obliquity of each scene surface with respect to the sensor. For example, text can be decoded more accurately when the normal of the surface on which it occurs is roughly perpendicular to the viewing direction. Furthermore, a SLAM-based approach can trace the reoccurrence of a particular text fragment in successive image frames. Multiple observations can be combined to improve accuracy, e.g. through the use either of super-resolution methods [24], [7] to reduce blur before OCR, or probabilistic lexical methods [17], [30] to combine the noisy low-level text fragments produced by OCR. The present study focuses on the latter method.

Some designers of text detection have used the texture-based Discrete Cosine Transform (DCT) to detect text in video [4], [8]. Others have used MSER, which is fast and

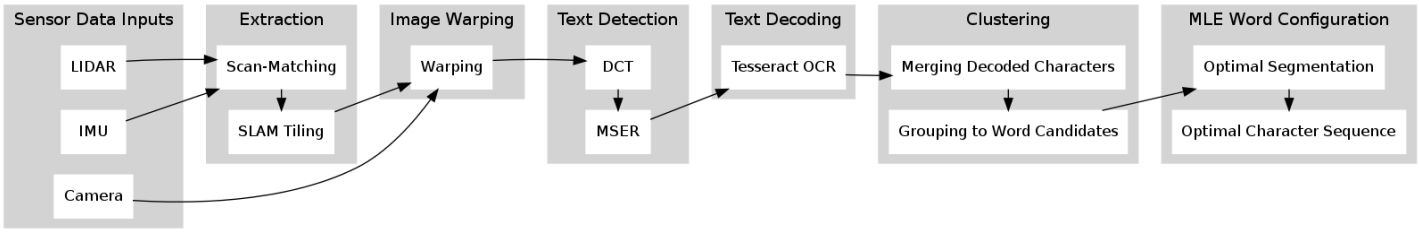

Fig. 4. The system workflow. Our system takes images and laser range data as inputs, extracts tiles, invokes text detection on tiles, and finally schedules text decoding for those tiles on which text was detected.

robust to blur, low contrast, and variation in illumination, color and texture [21]. We use an 8×8-pixel window DCT as a first-stage scan, then filter by size and aspect ratio. For blurry inputs, individual characters of a word usually merge into one connected component, which could be explored in the component tree generated by MSER [22], [20], [16]; see Fig. 3. We use MSER with shape descriptors for second-stage classification, to extract individual characters and to produce multiple detection regions for each character, which are then provided to Tesseract.

The availability of multiple observations of each tile enable our method to integrate information (Fig. 1). A clustering process groups decoded characters across multiple frames incorporating spatial separation and lexical distance. Candidate interpretations are combined within each group (representing a single character) using statistical voting with confidence scores. A second clustering step merges groups to form word candidates using another distance function.

Extracting environment text from word candidates is similar to the problem of handwriting word recognition, which involves (i) finding an optimal word configuration (segmentation) and (ii) finding an optimal text string. Our approach differs from that of Wang et al. [31], [30], who considered (i) and (ii) as a single problem of optimal word configuration using pictorial structure; we separate (i) and (ii) in order to reduce running time and increase control over the individual aggregation stages.

III. System

Fig. 4 shows an overview of our system's workflow.

A. Sensor Data Inputs

Data was collected from a wearable rig containing a Hokuyo UTM-30LX planar LIDAR, a Point Grey Bumblebee2 camera, and a Microstrain 3DM-GX3-25 IMU, shown in Fig 5. The IMU provides pitch and roll information. All sensor data was logged using the LCM (Lightweight Communications and Marshaling) [10] package.

B. Extraction

As the sensor suite moves through the environment, the system maintains an estimate of the sensor rig's motion using incremental LIDAR scan-matching [1] and builds a local map consisting of a collection of line segments (Fig. 2). Two line segments are merged if the difference of their slopes

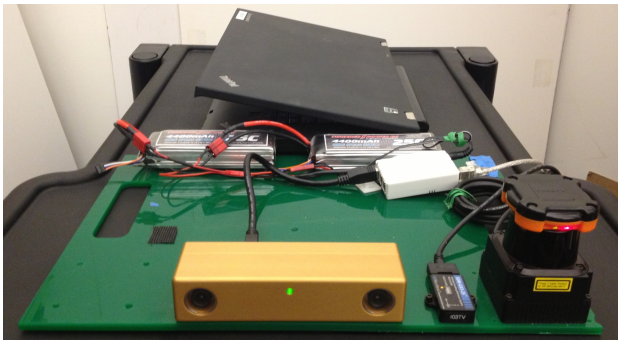

Fig. 5. The sensors were mounted on a rig and connected to a laptop computer for data collection.

is within a given threshold and offset. Each line segment is split into several 1-meter lateral extents which we call tiles. Newly visible tiles are added using the probabilistic Hough transform [2]. For each new tile the system creates four tile corners, each half a meter vertically and horizontally away from the tile center.

C. Image Warping

Any tiles generated within the field of view are then projected onto the frames of the cameras that observed them. Multiple observations can be gathered from various viewing positions and orientations. A fronto-parallel view of each tile is obtained for each observation through a homography transform constructed by generating a quadrilateral in OpenGL, and using projective texture mapping from the scene image onto the tile quadrilateral. A virtual camera is then placed in front of each tile to produce the desired fronto-parallel view of that tile at any desired resolution (we use 800×800 pixels). The per-tile transform is maintained, enabling later alignment of multiple observations in order to later improve image quality and OCR accuracy.

Each individual observation is associated with a tile (its unique identifier, corners, origin, and normal vector), the synthesized fronto-parallel image, and the camera pose. These observations are then passed to text detection and decoding.

D. Text Detection

The first stage of text detection applies an image pyramid to each tile in preparation for multi-scale DCT, with coefficients as per Crandall et al. [4]. The bounding box of each text

detection is then inspected using MSER [21] to extract shape descriptors, including aspect ratio and compactness. We set the MSER parameters as follows: aspect ratio less than 8, and compactness greater than 15. Scale-relevent parameters are estimated according to real-world setting (8 pixels per *cm*), corresponding to a minimum text height of 3 cm, and a minimum MSER region of 3 cm^2. The parameters for DCT detection include a minimum edge density of 8 edge-pixels per 8×8 window using Canny edge detection, with high and low hysteresis parameters equal to 100 and 200, respectively. For MSER detection, regions smaller than 5 pixels are discarded, and the parameter delta (the step size between intensity threshold levels) is set to 3 for higher sensitivity to blurry inputs. Both the DCT and MSER computations are implemented in OpenCV, with running times of about 10 msec and 300 msec, respectively.

E. Text Decoding

Decoding proceeds as follows. First, the image regions produced by either DCT or MSER (as gray-scale or binary images) are processed by the Tesseract OCR engine. Using the provided joined character chopping and broken character association, the binary inputs are segmented into one or multiple observations, i.e., the segmentation results from a MSER region. Tesseract outputs with too large an aspect ratio are removed. Each block is classified into a few candidates with confidence scores, for example, "B", "E" and "8" for the crop of an image of character "B." We set a minimum confidence score of 65 given by Tesseract to reduce incorrectly decoded characters. Running time depends on the number of input regions, but is usually less than 300 msec.

F. Clustering for Character and Word Candidates

A clustering module is used to: (a) merge decoded characters across multiple observations, and (b) cluster groups of decoded characters into word candidates. For (a), a distance predicate is implemented by Euclidean distance, text height, similarity between decoded results. Multiple observations can be obtained either across multiple frames or within a single frame. The parameters of multi-frame integration depend on system calibration. For (b), the confidence of groups of decoded characters, size of decoded characters, and Euclidean distance are applied. The confidence is determined by the number of decoded characters in the group; only groups with confidence above a threshold are selected. The threshold is $\sqrt{N_{obs}}/k$, where N_{obs} is the total number of accumulated decoded characters, and k is an arbitrary scalar. The bounding box of each decoded character in selected groups are overlaid on a density map, which is then segmented into regions. All selected groups of decoded characters are assigned to a region, representing a word candidate.

G. Finding Optimal Word Configuration and String

To extract whole words, we implemented a graph to combine spatial information (block overlaps). The output is a sequence of characters with each character comprising a small number of candidates provided by Tesseract. To recover the optimal word string each candidate from each group of decoded characters is considered as a node in a trellis, where the probability of each node arises from normalized voting using confidence scores. The prior probability is computed using bi-grams from an existing corpus [11]. We retain the top three candidates for each group of decoded characters, and use Viterbi's algorithm [26] for decoding. We seek an optimal character sequence W^*, as shown in Eq 1, where $P(Z|C_i)$ is the probability of nodes from the confidence-scored observations, and $P(C_i|C_{i-1})$ is the prior probability from the bi-gram.

$$W^* = \underset{w}{\operatorname{argmax}} \left(\sum P(Z|C_i) P(C_i|C_{i-1}) \right) \tag{1}$$

IV. Experimental Results

Text examples in public datasets (e.g. ICDAR and SVT) usually occur within high-quality (high-resolution, minimally blurry) imagery. In our setting, text often occurs within lower-resolution and much more blurried imagery. Our focus is to achieve text-spotting in a real-time system moving through an environment. We first examine how much the information about the surround given by SLAM and the warping process affect text detection and decoding in video frames. Next, we demonstrate the alignment of warped tile observations. Finally, we evaluate the accuracy gains arising from spatiotemporal fusion.

The evaluation is performed using a metric defined over m ground truth words and n decoded words. The $m \times n$ pairs of strings are compared using minimum edit distance d_{ij} for the i^{th} ground truth word and the j^{th} decoded word. A score S_{ij} for each pair is calculated as $(N_i - d_{ij})/N_i$, where N_i is the number of character of ground truth word i, when $N_i - d_{ij} > 0$, whereas S_{ij} is 0 otherwise. The accuracy is then measured by Eq 2, where the weight of each ground truth word w_i is set to $1/max(m, n)$ to penalize false alarms when $n > m$.

$$\text{Accuracy} = \sum_i w_i \max_j (S_{ij}) \tag{2}$$

A. Warping Accuracy with Distance and Obliquity

We mounted all equipment on a rig placed at waist height on a rolling cart, with the LIDAR sampling at 40 Hz and the camera sampling at 15 Hz. We attached signs with 140-point (5 cm) font at various wall locations. We pushed the cart slowly toward and by each sign to achieve varying view angles with respect to the sign's surface normal (Fig. 6(a) and Fig. 6(b)). The experiments were designed to evaluate text-spotting performance under varying viewing distance and obliquity, given that such factors effect the degree of bluriness in imagery.

Each original tile and its warped observation cropped from scene image frame was sent to Tesseract, our baseline decoder. Text spotting performance vs. the baseline is plotted as a function of viewing distance (Fig. 6(c) and Fig. 6(d)). Examples are shown in Fig. 6(e) and Fig. 6(f).

The results suggest that the baseline decoder works poorly when text is observed at distances greater than 1.5 m, and generally performs better for the warped observation than for the original ones. When the viewing direction is about 45 degrees to the surface normal, the accuracy of warping

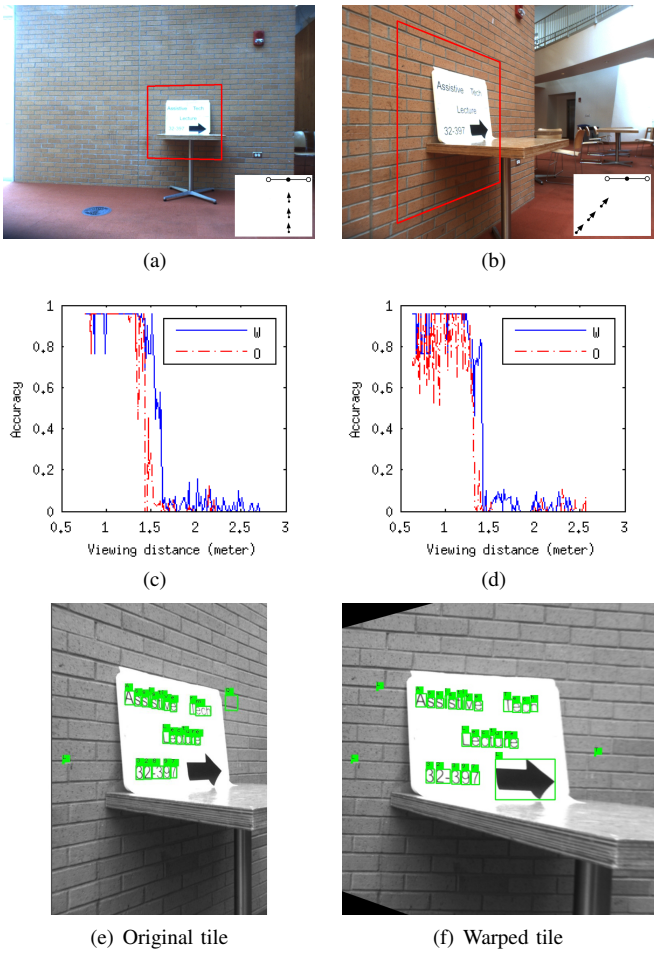

Fig. 6. Experiment settings and accuracy comparison of original and warped observations. (a) The normal of the surface is roughly antiparallel to the viewing direction. (b) The normal of the surface is about 45 degrees away from the viewing direction. Plots (c) and (d) show the accuracy of baseline decoding of original (O) and warped (W) tiles with respect to viewing distance for observations (a) and (b). (e) An original tile observation from 0.71 meters. (f) The warped observation corresponding to (e). The accuracy scores of (e) and (f) are 0.67 and 0.96, respectively.

observation is more consistent than that of original, which may be due to the skewed text line and perspective transformation of characters.

Given the limitation of baseline decoder, our proposed method intends to extend the capability of detecting and decoding more blurry imagery by spatiotemporal fusion of multiple observations. One key factor for integration is: how well are the warped observations aligned? we report the alignment and the calibration of sensors in the next section.

B. Alignment of Warped Observations

The distribution of decoded characters is shown in Fig. 7(a) and 7(b). Misalignment among certain decoded characters was measured manually. In 7(a), the logs were collected when the sensors were placed on a cart. The results suggest that the drift of decoded characters was uncertain to within about 20 pixels.

Another log was collected when the rig was hand-carried at about chest height by an observer who walked within an indoor environment. Fig. 7(b) demonstrates that imagery, to

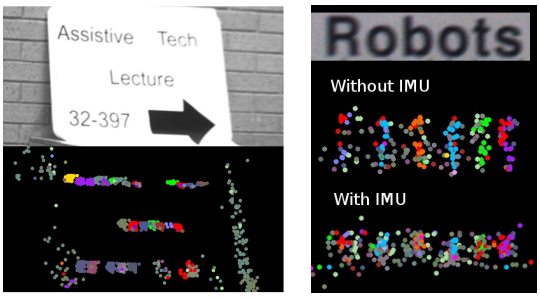

Fig. 7. The distribution of decoded characters. (a) There were only slight vertical and horizontal shifts. (b) Comparison between data with and without IMU for the second dataset (hand-carried). There were longer vertical drifts without IMU, but use of the IMU reduces drift.

	Single Frame	Multiple Frames
Merge decoded characters		
Euclidean distance	10	30
Text height scalar	2	2
Decoded text similarity	1	1
Group to word candidates		
Threshold of characters per group	1	$\sqrt{N_{obs}}/k$
Threshold parameter k		1.3
Size outlier scalar	5	2
Text height outlier scalar	5	2
Characters per word	3	3
Word aspect ratio min	1	1
Bounding box horizontal increment	0.3	0.3
Bounding box vertical increment	0.05	0.05

TABLE I. *Parameter settings for clustering decoded characters and word candidates.*

be aligned, required shifts of around 20 pixels horizontally and 70 pixels vertically without IMU data. When IMU data were integrated, the vertical shifts required reduced to around 35 pixels.

Given the alignment, we chose the experiment described in Fig. 6(b) to report the text-spotting performance of fusion of multiple observations. The parameter settings for clustering decoded characters and word candidates are shown in Table I. Comparing single and multiple frame integrations, Euclidean distance is the major factor for merging decoded characters, whereas the threshold of number of decoded character per group is the major factor for grouping to word candidates.

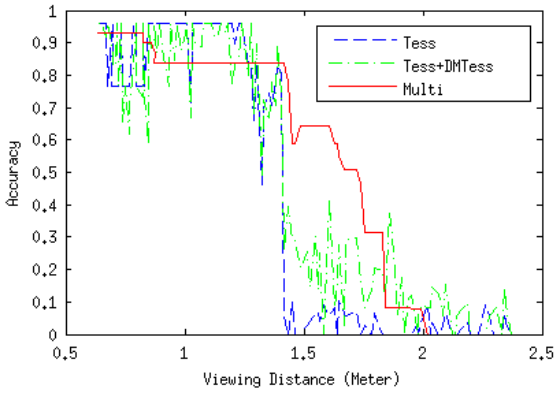

Fig. 8. Accuracy comparison with respect to viewing distance for observations.

C. Performance with Multiple Observations

We demonstrate that the proposed method combines noisy individual observations into a higher-confidence decoding. Fig. 8 plots the accuracy of (a) running Tesseract on the entire tile observation (Tess), (b) combining (a) and the proposed spotting pipeline into a single-frame detector (Tess+DMTess), and (c) fusing multiple observations from the proposed pipeline (Multi). The area under curve (AUC) values are 0.71, 0.79, and 0.91, respectively; these represent the overall performance of each spotting pipeline. The results suggest that Tess+DMTess moderately extends (to 2.4m from 1.5m) the distance at which text can be decoded, and Multi moderately improves the accuracy with which blurry text can be decoded (since blur tends to increase with viewing distance). We found that reducing the rate of false positives is critical to successful fusion process, because high false-alarm rate tends to cause our clustering method (§ III-F) to fail. We will continue to investigate our observation that Tess+DMTess outperforms Multi for close observations (1-1.5m).

V. Conclusion and Future Work

We described a SLAM-based text spotting method which detects and decodes scene text by isolating "tiles" arising from scene surfaces observed by a moving sensor suite. Such mode of operation poses challenges to conventional text detection for still imagery and stationary video frame. We demonstrate how information about the surroundings given by SLAM can be used to improve text spotting performance. We also show how to merge text extracted from multiple tile observations, yielding higher-confidence word recovery end-to-end. Our future work will 1) incorporate a more sophisticated tile orientation and camera motion model into the observation alignment, clustering, and language model; 2) collect large-scale datasets for evaluation; and 3) schedule computationally intensive inspection according to a spatial prior on text occurrence, thereby improving efficiency over the baseline method. Finally we plan to explore the use of the method to support task performance in robotics and assistive technology for blind and visually impaired people.

Acknowledgment

We thank the Andrea Bocelli Foundation for their support, and Javier Velez and Ben Mattinson for their contributions.

References

[1] A. Bachrach, S. Prentice, R. He, and N. Roy. RANGE - robust autonomous navigation in GPS-denied environments. *J. of Field Robotics*, 28(5):644–666, Sept. 2011.

[2] A. Bonci, T. Leo, and S. Longhi. A bayesian approach to the hough transform for line detection. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(6):945–955, 2005.

[3] X. Chen and A. Yuille. Detecting and reading text in natural scenes. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2004.

[4] D. Crandall, S. Antani, and R. Kasturi. Extraction of special effects caption text events from digital video. *International Journal on Document Analysis and Recognition*, 5(2-3):138–157, 2003.

[5] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2963–2970, 2010.

[6] M. F. Fallon, H. Johannsson, J. Brookshire, S. Teller, and J. J. Leonard. Sensor fusion for flexible human-portable building-scale mapping. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Algarve, Portugal, 2012.

[7] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *Image Processing, IEEE Transactions on*, 13(10):1327–1344, 2004.

[8] H. Goto. Redefining the dct-based feature for scene text detection. *International Journal of Document Analysis and Recognition (IJDAR)*, 11(1):1–8, 2008.

[9] X.-S. Hua, P. Yin, and H.-J. Zhang. Efficient video text recognition using multiple frame integration. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–397–II–400 vol.2, 2002.

[10] A. Huang, E. Olson, and D. Moore. LCM: Lightweight communications and marshalling. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, Oct. 2010.

[11] M. N. Jones and D. J. K. Mewhort. Case-sensitive letter and bigram frequency counts from large-scale english corpora. *Behavior Research Methods, Instruments, & Computers*, 36(3):388–396, 2004.

[12] K. Jung, K. I. Kim, and A. K. Jain. Text information extraction in images and video: a survey. *Pattern Recognition*, 37(5):977 – 997, 2004.

[13] H. Li and D. Doermann. Text enhancement in digital video using multiple frame integration. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 19–22, 1999.

[14] S. Lucas. Icdar 2005 text locating competition results. In *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR)*, pages 80–84 Vol. 1, 2005.

[15] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004.

[16] C. Merino-Gracia, K. Lenc, and M. Mirmehdi. A head-mounted device for recognizing text in natural scenes. In *Proc. of Camera-based Document Analysis and Recognition (CBDAR)*, pages 29–41, 2011.

[17] A. Mishra, K. Alahari, and C. Jawahar. Top-down and bottom-up cues for scene text recognition. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2687–2694. IEEE, 2012.

[18] G. K. Myers and B. Burns. A robust method for tracking scene text in video imagery. *CBDAR05*, 2005.

[19] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Asian Conf. on Computer Vision (ACCV)*, pages 770–783, 2004.

[20] L. Neumann and J. Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR)*, pages 687–691, 2011.

[21] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[22] D. Nister and H. Stewenius. Linear time maximally stable extremal regions. In *Eur. Conf. on Computer Vision (ECCV)*, pages 183–196, 2008.

[23] E. Olson. Real-time correlative scan matching. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4387–4393, Kobe, Japan, June 2009.

[24] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, 20(3):21–36, 2003.

[25] I. Posner, P. Corke, and P. Newman. Using text-spotting to query the world. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3181–3186, 2010.

[26] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[27] T. Sato, T. Kanade, E. Hughes, and M. Smith. Video OCR for digital news archive. In *Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on*, pages 52–60, 1998.

[28] R. Smith. An overview of the tesseract OCR engine. In *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR)*, page 629633, 2007.

[29] R. Smith. History of the Tesseract OCR engine: what worked and what didn't. In *Proc. of SPIE Document Recognition and Retrieval*, 2013.

[30] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Intl. Conf. on Computer Vision (ICCV)*, 2011.

[31] K. Wang and S. Belongie. Word spotting in the wild. In *Eur. Conf. on Computer Vision (ECCV)*, 2010.

[32] C. Yi and Y. Tian. Assistive text reading from complex background for blind persons. In *Proc. of Camera-based Document Analysis and Recognition (CBDAR)*, pages 15–28, 2011.